What is claimed is:

1. A method of improving installation of software packages, comprising steps of:

defining an object model representing a plurality of components of a software installation process, wherein the defined model enables specifying conditional installation information for the components; and

populating the object model to describe a particular software installation package, wherein the conditional installation information is populated with information to describe conditional installation scenarios.

2. The method according to Claim 1, further comprising steps of:

using the conditional installation information of the populated object model during an installation of the particular software installation package to determine whether the installation should be performed; and

performing the installation if so.

3. The method according to Claim 1, further comprising the step of instantiating a plurality of objects according to the defined object mode, each of the instantiated objects corresponding to a selected one of the components of the software installation process, and wherein the populating step populates the instantiated objects.

4. The method according to Claim 3, wherein the instantiated objects are JavaBeans.

5. The method according to Claim 1, wherein the conditional installation information comprises a suite-level conditional installation component.

6. The method according to Claim 1, wherein the conditional installation information comprises one or more software component-level conditional installation components.

7. The method according to Claim 1, wherein the conditional installation information comprises a suite-level conditional installation component and one or more software component-level conditional installation components.

8. The method according to Claim 7, further comprising the step of evaluating the suite-level conditional installation component and/or the one or more software component-level conditional installation components as preconditions to installing a corresponding one of the components of the particular software installation package.

9. The method according to Claim 7, further comprising the step of evaluating the suite-level conditional installation component and/or the one or more software component-level conditional installation components as preconditions to downloading and installing a corresponding one of the components of the particular software installation package.

10. The method according to Claim 1, wherein the conditional installation information comprises an executable code module.

11. The method according to Claim 1, wherein the conditional installation information comprises a reference to an executable code module.

12. The method according to Claim 3, further comprising the step of caching one or more of the plurality of instantiated objects.

13. The method according to Claim 12, wherein the caching step further comprises the step of avoiding downloading of the cached ones of the plurality of instantiated objects.

14. A system for improving installation of software packages, comprising:

means for defining an object model representing a plurality of components of a software installation process, wherein the defined model enables specifying conditional installation information for the components; and

means for populating the object model to describe a particular software installation package, wherein the conditional installation information is populated with information to describe conditional installation scenarios.

15. The system according to Claim 14, further comprising:

means for using the conditional installation information of the populated object model during an installation of the particular software installation package to determine whether the installation should be performed; and

means for performing the installation if so.

16. The system according to Claim 14, further comprising means for instantiating a plurality of objects according to the defined object mode, each of the instantiated objects corresponding to a selected one of the components of the software installation process, and wherein the means for populating populates the instantiated objects.

17. The system according to Claim 16, wherein the instantiated objects are objects in a scripting language.

18. The system according to Claim 1, wherein the conditional installation information comprises a suite-level conditional installation component and one or more software component-level conditional installation components.

19. The system according to Claim 18, further comprising means for evaluating the suite-level conditional installation component and/or the one or more software component-level conditional installation components as preconditions to downloading and installing a corresponding one of the components of the particular software installation package.

20. The system according to Claim 14, wherein the conditional installation information comprises an executable code module.

21. The system according to Claim 14, wherein the conditional installation information comprises a reference to an executable code module.

22. The system according to Claim 16, further comprising means for caching one or more of the plurality of instantiated objects.

23. The system according to Claim 22, wherein the means for caching further comprises means for avoiding downloading of the cached ones of the plurality of instantiated objects.

24. A computer program product for improving installation of software packages, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for defining an object model representing a plurality of components of a software installation process, wherein the defined model enables specifying conditional installation information for the components; and

computer-readable program code means for populating the object model to describe a particular software installation package, wherein the conditional installation information is populated with information to describe conditional installation scenarios.

25. The computer program product according to Claim 24, further comprising:

computer-readable program code means for using the conditional installation information of the populated object model during an installation of the particular software installation package to determine whether the installation should be performed; and

computer-readable program code means for performing the installation if so.

26. The computer program product according to Claim 25, further comprising computer-readable program code means for instantiating a plurality of objects according to the defined object mode, each of the instantiated objects corresponding to a selected one of the components of the software installation process, and wherein the computer-readable program code means for populating populates the instantiated objects.

27. The computer program product according to Claim 26, wherein the instantiated objects are structured documents

28. The computer program product according to Claim 24, wherein the conditional installation information comprises a suite-level conditional installation component and one or more software component-level conditional installation components.

29. The computer program product according to Claim 28, further comprising computer-readable program code means for evaluating the suite-level conditional installation component and/or the one or more software component-level conditional installation components as preconditions to installing a corresponding one of the components of the particular software installation package.

30. The computer program product according to Claim 29, further comprising computer-

readable program code means for evaluating the suite-level conditional installation component and/or the one or more software component-level conditional installation components as preconditions to installing a corresponding one of the components of the particular software installation package.

31. The computer program product according to Claim 24, wherein the conditional installation information comprises an executable code module.

32. The computer program product according to Claim 26, further comprising computer-readable program code means for caching one or more of the plurality of components.

33. The computer program product according to Claim 32, wherein the computer-readable program code means for caching further comprises computer-readable program code means for avoiding downloading of the cached ones of the plurality of components.